

Settima esercitazione

Shell scripting

Agenda

Esempio

Esplorazione completa di una directory: script bash con ricorsione.

Esercizio 1

Esplorazione ricorsiva del file system

Esercizio 2

Esempio di compito d'esame

Esempio – Script ricorsivi

Si scriva uno script bash avente interfaccia di invocazione

reurse_dir.sh dir

Il programma, dato un directory in ingresso **dir**, deve stampare su stdout l'elenco dei file contenuti nel directory e in tutti i suoi sottodirectory (analogamente al comando **ls -R**)

Schema di soluzione ricorsiva

`recurse_dir.sh arg1`

caso **base**

arg1 è un file

→ stampo il nome

caso generale espresso in termini **ricorsivi**

arg1 è una directory

→ mi muovo nella directory **arg1**

per ogni file (normale o directory) **invoco**

nuovamente recurse_dir.sh

Bozza di soluzione

```
#!/bin/bash
if ! test -d "$1" ; then
    echo `pwd`/$1
```

Caso
base

```
else
    cd "$1"
    for f in * ; do
        "$0" "$f"
    done
fi
```

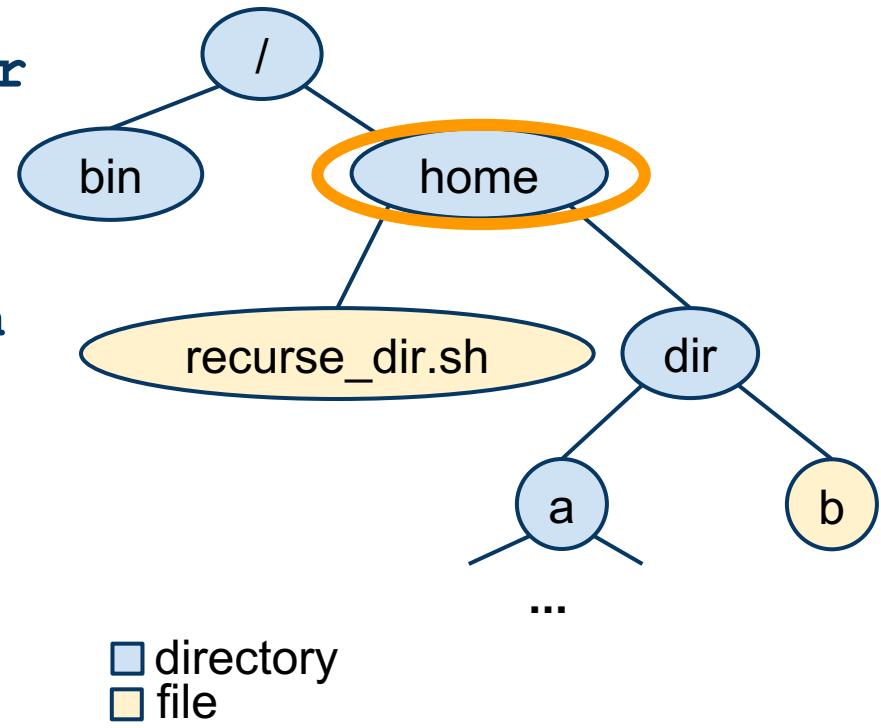
Caso
generale

Chiamata ricorsiva

Ricorsione (1/6)

```
$ pwd  
/home  
$ /home/recurse_dir.sh dir
```

```
if ! test -d "$1" ; then  
    echo `pwd`/$1  
else  
    cd "$1"  
    for f in * ; do  
        "$0" "$f"  
    done  
fi
```



VARIABILI:

\$PWD /home

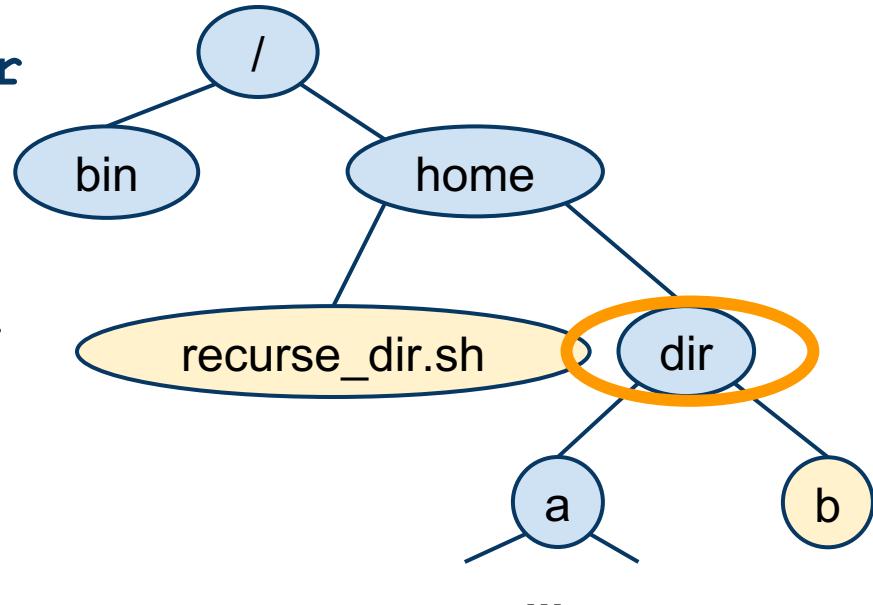
\$0 /home/recurse_dir.sh

\$1 dir

Ricorsione (2/6)

```
$ pwd  
/home  
$ /home/recurse_dir.sh dir
```

```
if ! test -d "$1" ; then  
    echo `pwd`/$1  
else  
    → cd "$1"  
    for f in * ; do  
        "$0" "$f"  
    done  
fi
```



■ directory
■ file

VARIABILI:

\$PWD /home/dir

\$0 /home/recurse_dir.sh

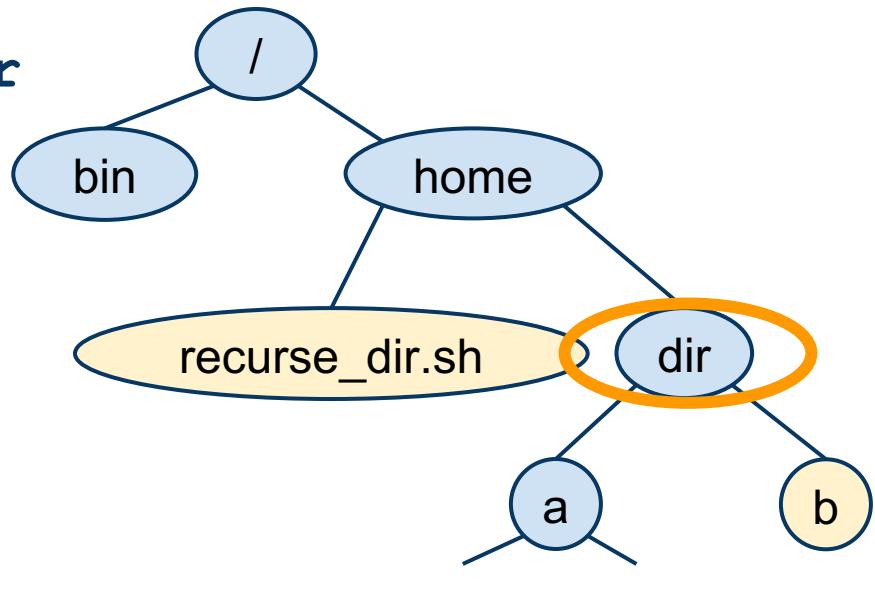
\$1 dir

Ricorsione (3/6)

```
$ pwd  
/home  
$ /home/recurse_dir.sh dir
```

```
$ /home/recuse_dir.sh a
```

```
if ! test -d "$1" ; then  
    echo `pwd`/$1  
else  
    cd "$1"  
    for f in * ; do  
        echo "$0" "$f"  
    done  
fi
```



- [blue square] directory
- [yellow square] file

VARIABILI:

\$PWD /home/dir

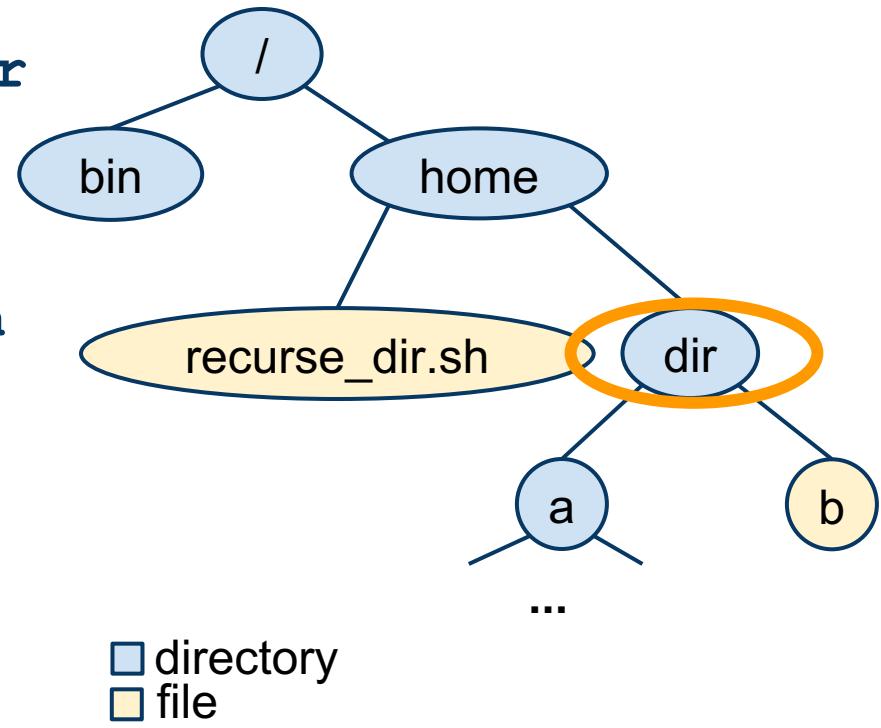
\$0 /home/recuse_dir.sh

\$1 dir

Ricorsione (4/6)

```
$ pwd  
/home  
$ /home/recurse_dir.sh dir
```

```
if ! test -d "$1" ; then  
    echo `pwd`/$1  
else  
    cd "$1"  
    for f in * ; do  
        "$0" "$f"  
    done  
fi
```



VARIABILI:

\$PWD /home/dir

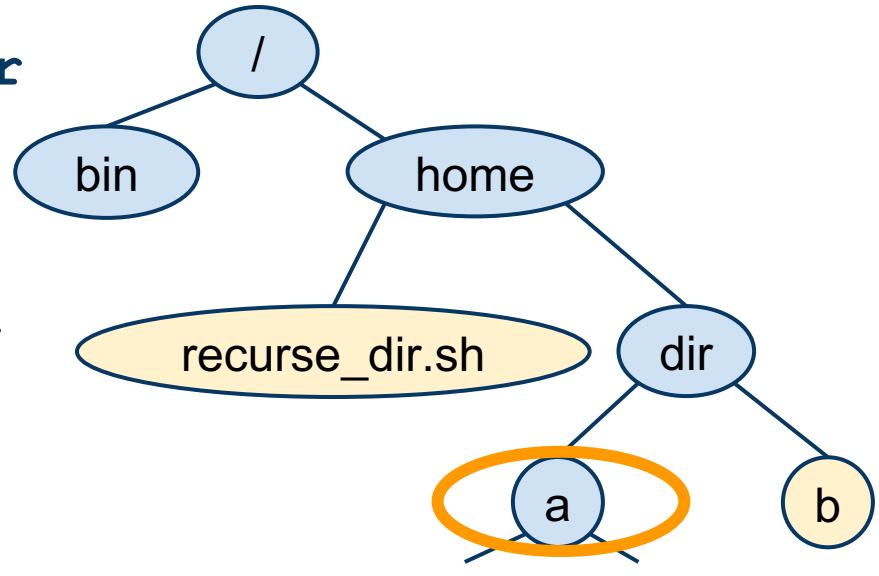
\$0 /home/recurse_dir.sh

\$1 a

Ricorsione (5/6)

```
$ pwd  
/home  
$ /home/recurse_dir.sh dir
```

```
if ! test -d "$1" ; then  
    echo `pwd`/$1  
else  
    cd "$1"  
    for f in * ; do  
        "$0" "$f"  
    done  
fi
```



■ directory
■ file

VARIABILI:

\$PWD /home/dir/a

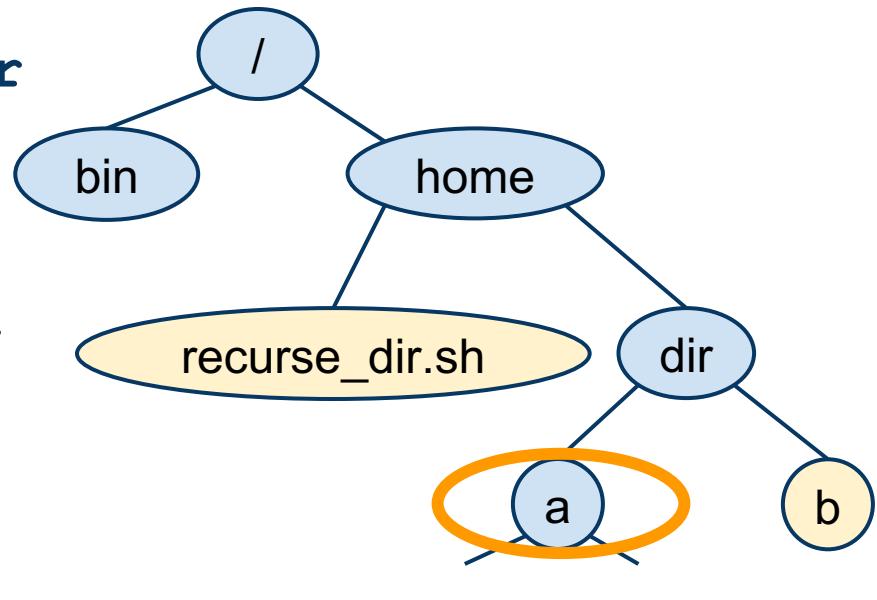
\$0 /home/recurse_dir.sh

\$1 a

Ricorsione (6/6)

```
$ pwd  
/home  
$ /home/recurse_dir.sh dir
```

```
if ! test -d "$1" ; then  
    echo `pwd`/$1  
else  
    cd "$1"  
    for f in * ; do  
        "$0" "$f"  
    done  
fi
```



- directory
- file

VARIABILI:

\$PWD /home/dir/a
\$0 /home/recurse_dir.sh
\$1 ...

ATTENZIONE

Nell'esempio lo script è stato invocato specificando il suo path assoluto:

```
$ /home/recurse_dir.sh dir
```

Cosa succederebbe invocandolo
con un path relativo?

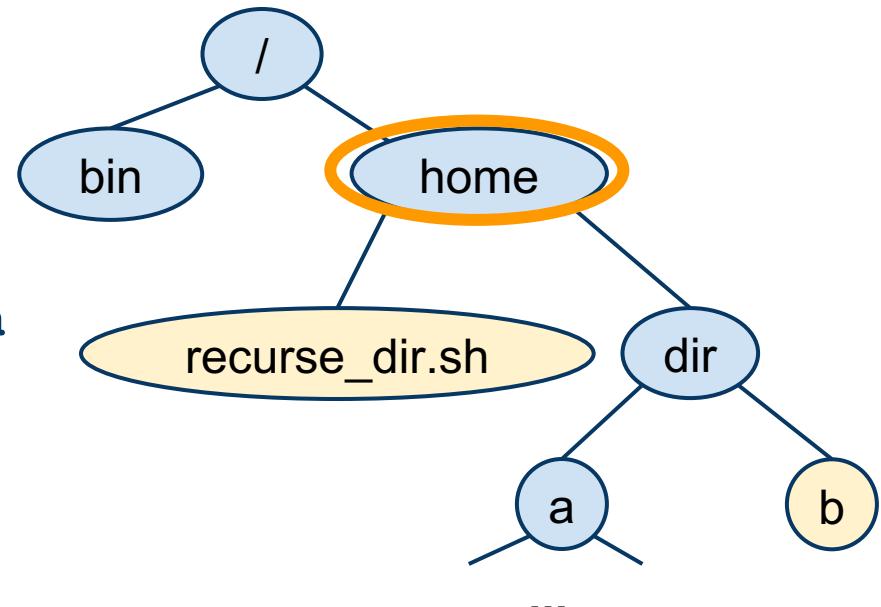
```
$ ./recurse_dir.sh dir
```



Ricorsione - alternativa (1/3)

```
$ pwd  
/home  
$ ./recurse_dir.sh dir
```

```
if ! test -d "$1" ; then  
    echo `pwd`/$1  
else  
    cd "$1"  
    for f in * ; do  
        "$0" "$f"  
    done  
fi
```



- directory
- file

VARIABILI:

\$PWD /home

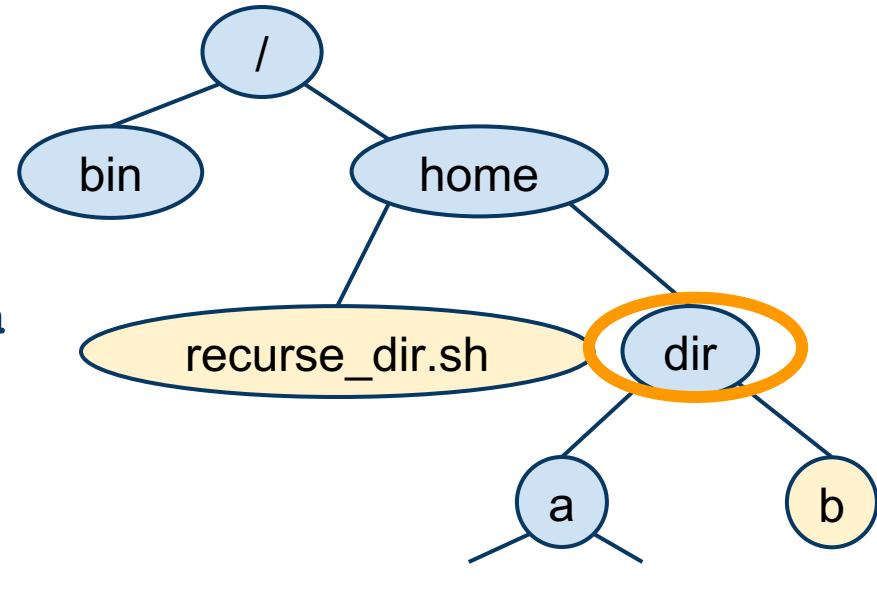
\$0 ./recurse_dir.sh

\$1 dir

Ricorsione - alternativa (2/3)

```
$ pwd  
/home  
$ ./re recurse_dir.sh dir
```

```
if ! test -d "$1" ; then  
    echo `pwd`/$1  
else  
    → cd "$1"  
    for f in * ; do  
        "$0" "$f"  
    done  
fi
```



■ directory
■ file

VARIABILI:

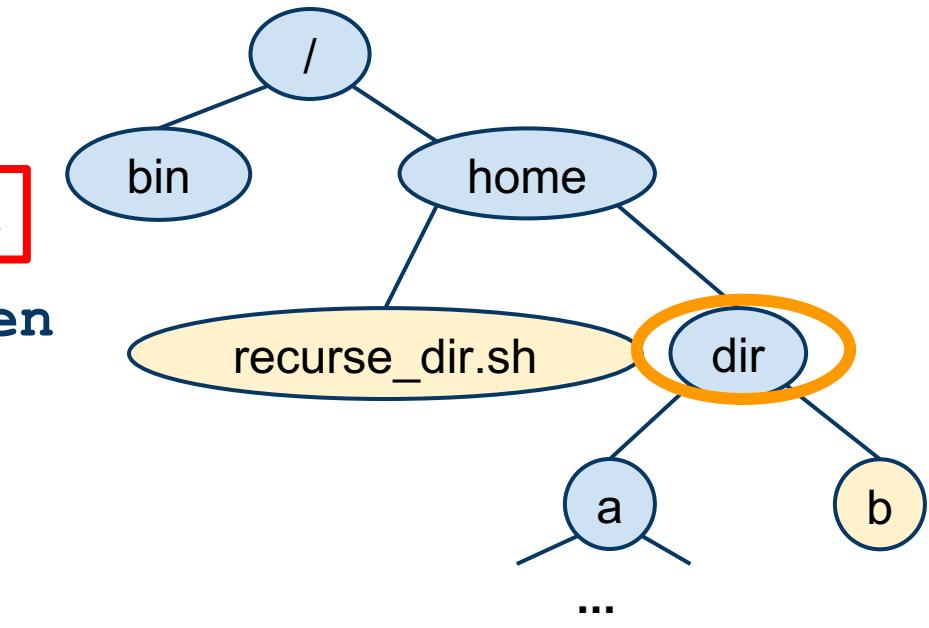
\$PWD /home/dir
\$0 ./re recurse_dir.sh
\$1 dir

Ricorsione - alternativa (3/3)

```
$ pwd  
/home  
$ ./recurse_dir.sh dir
```

```
$ ./recurse_dir.sh a
```

```
if ! test -d "$1" ; then  
    echo $PWD/$1  
else  
    cd "$1"  
    for f in * ; do  
        "$0" "$f"  
    done  
fi
```



■ directory
■ file

VARIABILI:

\$PWD /home/dir

\$0 ./recurse_dir.sh

\$1 dir

Come risolvere?

Problema: Un valore dipendente dalla directory di lavoro corrente (un percorso relativo) viene "propagato" da una invocazione ricorsiva all'altra (tramite la variabile **\$0**)
La directory di lavoro però cambia (perchè usiamo il comando **cd** nel codice)

Possibile soluzione: Prima di iniziare la ricorsione memorizzare la directory di partenza in una variabile che verrà usata per le invocazioni ricorsive

Occorre creare:

- **Script ricorsivo**
- **Script di invocazione:**

Controlla i parametri

Salva in maniera "stabile" il percorso dello script ricorsivo
Innesca la ricorsione

Struttura di un file comandi ricorsivo

invoker.sh

`#!/bin/sh`

Controllo degli argomenti

Invocazione del file comandi ricorsivo `do_recursive.sh`

`do_recursive.sh`

`#!/bin/sh`

Esecuzione del compito

Invocazione del file comandi ricorsivo `do_recursive.sh`

Script di invocazione

re recurse_dir.sh

```
#!/bin/bash
# ... controllo argomenti

oldpath=$PATH
PATH=$PATH:`pwd`  
do_recurse_dir.sh "$1"
PATH=$oldpath
```

do_recurse_dir.sh

```
#!/bin/bash
if ! test -d "$1" ; then
    echo `pwd`/$1
else
    cd "$1"
    for f in * ; do
        "$0" "$f"
    done
fi
```

PATH è una variabile d'ambiente che contiene dei path di directory separate da ":".

Quando lancio un comando senza alcun path (né assoluto né relativo, es: invoco **ls** invece di **/usr/bin/ls**), il SO cerca quel comando in tutte le directory contenute nella variabile **PATH**.

Script di invocazione

`recurse_dir.sh`

```
#!/bin/bash
# ... controllo argomenti

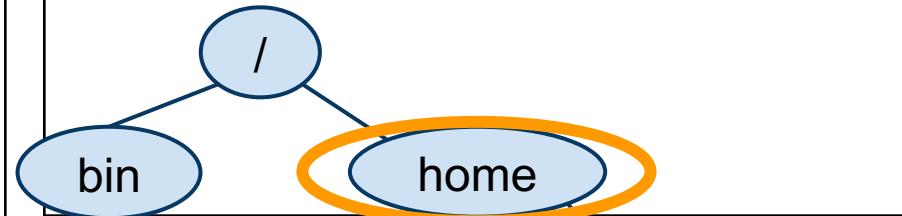
oldpath=$PATH
PATH=$PATH:`pwd`  

do_recurse_dir.sh "$1"
PATH=$oldpath
```

`do_recurse_dir.sh`

```
#!/bin/bash
```

```
if ...
```



Problema :

Che succede se gli script si trovano in
/home/dloreti e l'utente li invoca dalla directory corrente
/home con il path relativo: **./dloreti/recurse_dir.sh**
=> `pwd` viene espanso in “**/home**”
=> **PATH=\$PATH:/home**
=> **do_recurse_dir.sh** viene cercato in **/home** ...NON TROVATO!

Soluzione generale

recurse_dir.sh

```
#!/bin/bash
# ... controllo argomenti

if [[ "$0" = /* ]] ; then
    # se $0 è un path assoluto
    dir_name=`dirname "$0"`
    recursive_cmd="$dir_name/do_recurse_dir.sh"
elif [[ "$0" = */* ]] ; then
    # se c'è uno slash, ma non inizia con /
    # $0 è un path relativo
    dir_name=`dirname "$0"`
    recursive_cmd=`pwd`/$dir_name/do_recurse_dir.sh
else
    # Non si tratta né di un path relativo, né di uno
    # assoluto, il comando $0 sarà cercato in $PATH.
    recursive_cmd=do_recurse_dir.sh
fi
#Invoco il comando ricorsivo
"$recursive_cmd" "$1"
```

Restituisce \$0 tranne
l'ultimo / e ciò che segue

do_recurse_dir.sh

```
#!/bin/bash
if ...
```

Esercizio 1 – Esplorazione ricorsiva del file system (1/2)

Realizzare un file comandi (ricorsivo) che abbia la sintassi

`print_heads size dir`

dove:

`size` è un intero

`dir` è il nome **assoluto** di un direttorio esistente nel file system

Esercizio 1 - (2/2)

Il compito del file comandi è quello di :

- Visitare (ricorsivamente) il sottoalbero individuato da **dir**
- Individuare i regular file la cui **dimensione in KB** sia superiore a **size**
- Per ogni file che rispetti tale caratteristica stampare a video il suo nome assoluto e i primi 10 caratteri in esso contenuti

Esercizio 1 : suggerimenti

Prima di tutto realizzare la ricorsione e testare che funzioni correttamente!

Poi:

- Individuare i file con **dimensione in KB** superiore a **size** → vedere il comando **stat** per ottenere la dimensione del file e la sua rappresentazione in KB (**man stat**)
- Stampare a video i primi 10 caratteri nel file → vedere il comando **head** con opzione **-c** per ottenere i byte di “testa” del file (**man head**)

Esercizio 2 - Esplorazione ricorsiva di N directory (1/2)

Realizzare un file comandi (ricorsivo) che abbia la sintassi

print_heads size dir1...dirN

dove:

size è un intero

dir1...dirN sono nomi **assoluti** di direttori esistenti nel file system

Esercizio 2 - (2/2)

Il compito del file comandi è quello di :

- Visitare (ricorsivamente) ogni sottoalbero individuato da **dir1...dirN**
- Individuare i regular file la cui **dimensione in KB** sia superiore a **size**
- Per ogni file che rispetti tale caratteristica stampare a video il suo nome assoluto e i primi 10 caratteri in esso contenuti
- Usando il comando **tee -a**, stampare tale informazione anche su un file **riepilogo.txt** nella home dell'utente che ha invocato lo script